1 Iterative Solution of Elliptic PDEs

In the previous exercises, we have worked with partial differential equations (PDEs) known as evolution equations. They are called evolution equations because of time dependent terms of the form $\frac{\partial u}{\partial t}$. These partial derivatives with respect to time allow the PDEs to change or evolve in time. To numerically solve an evolution equation, it is necessary to advance a given initial state forward in time. For the current exercise, we will consider non-evolution equations such as the Laplace equation $u_{xx}+u_{yy}=0$ and the Poisson equation $u_{xx}+u_{yy}=f(x,y)$. Notice the absence of any explicit time dependence. Problems of this type are also known as *elliptic boundary value problems*. A *boundary value problem* (BVP) is a PDE that has no initial condition, but rather must satisfy a set condition at the boundary.

Lets consider a 1-D toy example posed on the interval $0 \leq x \leq L$,

$$u_{xx} = f(x)\,, \quad u(0) = G \text{ and } u(L) = H. \tag{1}$$

The discrete problem is constructed by the following steps:

1. Construct a uniform grid of $N+1$ points covering $0 \leq x \leq L$. Let $\Delta x = L/N$, then $x_i = (iL)/N$ for $i = 0, \ldots, N$.

2. Replace the second derivative in (1) with the standard second order centered difference, where $u_i = u(x_i)$ and $f_i = f(x_i)$, then

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = f_i \quad i = 1, \ldots, N-1. \tag{2}$$

   Notice that the limits of (2) exclude the boundary points x_0 and x_N. This is because the boundary points, by definition, are fixed; $u_0 = G$ and $u_N = H$.

As an illustration of (2), let $N = 5$, then

$$\begin{aligned} u_2 - 2u_1 + u_0 &= \Delta x^2 f_1 \\ u_3 - 2u_2 + u_1 &= \Delta x^2 f_2 \\ u_4 - 2u_3 + u_2 &= \Delta x^2 f_3 \\ u_5 - 2u_4 + u_3 &= \Delta x^2 f_4 \end{aligned} \tag{3}$$

The system of equations in (3) can be recast as a linear matrix problem of the form $\mathbb{A}\mathbf{u} = \mathbf{f}$. Explicitly this is

$$\frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} (f_1 - G) \\ (f_2) \\ (f_3) \\ (f_4 - H) \end{bmatrix} \tag{4}$$

To solve for $\mathbf{u}$, the matrix ($\mathbb{A}$), on the left side of (4), must to be inverted. As N increases, the matrix equation (4) becomes very sparse. Because of this, a direct inversion of the matrix $\mathbb{A}$ is very inefficient. The preferred method is an iterative solver.

1.1 Iterative Solvers

An iterative matrix solver is one in which a first approximation of the solution is used to calculate the the second and subsequent approximations. The iterative procedure is said to converge if the difference between consecutive approximations diminishes.

Consider the example in (4)

$$\begin{aligned} u_2 - 2u_1 &= \Delta x^2(f_1) - G \\ u_3 - 2u_2 + u_1 &= \Delta x^2(f_2) \\ u_4 - 2u_3 + u_2 &= \Delta x^2(f_3) \\ -2u_4 + u_3 &= \Delta x^2(f_4) - H \end{aligned} \tag{5}$$

Taking each row of (5), and solving for the u of that row provides

$$\begin{aligned} u_1 &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2 - G\big) \\ u_2 &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1 - u_3\big) \\ u_3 &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2 - u_4\big) \\ u_4 &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3 - H\big) \end{aligned} \tag{6}$$

Jacobi Method: Denote each approximation by a superscript, thus the first approximation of u_1 is $u_1^{(1)}$, the second is $u_1^{(2)}$, and the k^{th} is $u_1^{(k)}$. Therefore the first iteration is

$$\begin{aligned} u_1^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2 - G\big) \\ u_2^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1 - u_3\big) \\ u_3^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2 - u_4\big) \\ u_4^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3 - H\big) \end{aligned} \tag{7}$$

where u_i (with no superscript) is the first guess needed to initialize the iterative scheme. The second iteration is

$$\begin{aligned} u_1^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2^{(1)} - G\big) \\ u_2^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1^{(1)} - u_3^{(1)}\big) \\ u_3^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2^{(1)} - u_4^{(1)}\big) \\ u_4^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3^{(1)} - H\big) \end{aligned} \tag{8}$$

Notice that the right hand side of the second iteration depends only on first iteration terms. The $k{+}1^{\text{th}}$ iteration is

$$\begin{aligned} u_1^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2^{(k)} - G\big) \\ u_2^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1^{(k)} - u_3^{(k)}\big) \\ u_3^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2^{(k)} - u_4^{(k)}\big) \\ u_4^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3^{(k)} - H\big) \end{aligned} \tag{9}$$

Gauss-Seidel Method: In this method the new iterate values are used as they become available. In the Gauss-Seidel method, the right hand side of the equations depends on both the current and previous iterates. This differs from the Jacobi method where all of the terms on right hand side are of the previous iterate. The first Gauss-Seidel iterate is

$$\begin{aligned} u_1^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2 - G\big) \\ u_2^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1^{(1)} - u_3\big) \\ u_3^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2^{(1)} - u_4\big) \\ u_4^{(1)} &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3^{(1)} - H\big) \end{aligned} \tag{10}$$

The second iteration is

$$\begin{aligned} u_1^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2^{(1)} - G\big) \\ u_2^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1^{(2)} - u_3^{(1)}\big) \\ u_3^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2^{(2)} - u_4^{(1)}\big) \\ u_4^{(2)} &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3^{(2)} - H\big) \end{aligned} \tag{11}$$

and the general $k{+}1^{\text{th}}$ iterate

$$\begin{aligned} u_1^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_1 - u_2^{(k)} - G\big) \\ u_2^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_2 - u_1^{(k+1)} - u_3^{(k)}\big) \\ u_3^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_3 - u_2^{(k+1)} - u_4^{(k)}\big) \\ u_4^{(k+1)} &= -\frac{1}{2}\big(\Delta x^2 f_4 - u_3^{(k+1)} - H\big) \end{aligned} \tag{12}$$

Notice that the terms on the right hand side are both k^{th} and $k{+}1^{\text{th}}$ iterates. The first equation contains only terms from the k^{th} iterate. The second equation contains one term, the first term, from the new $k{+}1^{\text{th}}$ iterate. The remaining terms are from the k^{th} iterate. The third equation contains two terms, the first and second, from the new $k{+}1^{\text{th}}$ iterate. The remaining terms are from the k^{th} iterate. The i^{th} equation contains $i{-}1^{\text{th}}$ terms, the first $i-1$ terms, from the new $k{+}1^{\text{th}}$ iterate. The remaining terms are from the k^{th} iterate. For the general case of M equations,

$$u_i^{(k+1)} = -\frac{1}{2}\left(\Delta x f_i - \sum_{j=1}^{i-1} a_{ij} u_j^{(k+1)} - \sum_{j=i+1}^{M} a_{ij} u_j^{(k)}\right) \tag{13}$$

where the general coefficients a_{ij} have replaced the explicit values.

Successive over-relaxation Method (SOR) Add and subtract $u_i^{(k)}$ to the right side of the i^{th} Gauss-Seidel equation (12). The result looks like

$$\begin{aligned}
u_1^{(k+1)} &= u_1^{(k)} - \frac{1}{2}\left[\Delta x^2 f_1 + 2u_1^{(k)} - u_2^{(k)} - G\right] \\
u_2^{(k+1)} &= u_2^{(k)} - \frac{1}{2}\left[\Delta x^2 f_2 + 2u_2^{(k)} - u_1^{(k+1)} - u_3^{(k)}\right] \\
u_3^{(k+1)} &= u_3^{(k)} - \frac{1}{2}\left[\Delta x^2 f_3 + 2u_3^{(k)} - u_2^{(k+1)} - u_4^{(k)}\right] \\
u_4^{(k+1)} &= u_4^{(k)} - \frac{1}{2}\left[\Delta x^2 f_4 + 2u_4^{(k)} - u_3^{(k+1)} - H\right]
\end{aligned} \tag{14}$$

The expressions within the square brackets are corrections to $u_i^{(k)}$. Under certain circumstances, convergence can be accelerated by increasing the size of the correction term. This is the difference between the SOR and Gauss-Seidel methods. Consider an acceleration parameter (often called a relaxation factor) ω, which is in the range $1 < \omega < 2$. Then the SOR iteration is defined as

$$\begin{aligned}
u_1^{(k+1)} &= u_1^{(k)} - \frac{\omega}{2}\left[\Delta x^2 f_1 - G + 2u_1^{(k)} - u_2^{(k)}\right] \\
u_2^{(k+1)} &= u_2^{(k)} - \frac{\omega}{2}\left[\Delta x^2 f_2 - u_1^{(k+1)} + 2u_2^{(k)} - u_3^{(k)}\right] \\
u_3^{(k+1)} &= u_3^{(k)} - \frac{\omega}{2}\left[\Delta x^2 f_3 - u_2^{(k+1)} + 2u_3^{(k)} - u_4^{(k)}\right] \\
u_4^{(k+1)} &= u_4^{(k)} - \frac{\omega}{2}\left[\Delta x^2 f_4 - u_3^{(k+1)} + 2u_4^{(k)} - H\right]
\end{aligned} \tag{15}$$

The value of $\omega = 1$ gives the Gauss-Seidel iteration. The optimal value of ω is a function of the the PDE and the grid spacing, but not the boundary conditions. A common strategy is to search for the optimal ω by trial and error for a case where the boundary conditions allow an analytical solution.

For the general case of M equations,

$$u_i^{(k+1)} = u_i^{(k)} + \frac{\omega}{a_{ii}} \left(\Delta x f_i - \sum_{j=1}^{i-1} a_{ij} u_j^{(k+1)} - \sum_{j=i}^{M} a_{ij} u_j^{(k)} \right), i = 1, \ldots, M$$

1.2 Stopping Criteria

The final piece of the iterative solver is to define the stopping criteria. The stopping criteria determine when enough iterations have been completed. Ideally, two consecutive iterates would be compared. If their maximum difference decreases, the method converges. For a convergent method, it is sufficient to iterate until the maximum difference is below a threshold value.